A knowledge based method for the medical question answering problem

Rafael M. Terol *, Patricio Martínez-Barco, Manuel Palomar

Department of Software and Computing Systems; The University of Alicante, San Vicente del Raspeig Road, Alicante, Spain

Abstract

In this paper, a restricted domain Question Answering (QA) system is described. The design architecture of this QA system and the features that allow the adaptation of the QA system to the medical domain are also presented. The advantages of this QA system include the simple process of defining the question taxonomy answered by the system as well as the possibility of locally or remotely managed document collections. The main computing methods of the QA system are based on the application of Natural Language Processing (NLP) techniques to infer the logic forms and on the treatment of the logic forms. The knowledge of the system is acquired through the use of two different resources: Unified Medical Language System (UMLS) to handle the medical terminology and WordNet to manage the open-domain terminology.

Key words: Bioinformatics, Biomedical, Text mining, Medicine, Medinformatics, Question Answering Framework, Medical Question Taxonomies

1 Introduction

Open-domain textual Question-Answering (QA), as defined by the TREC competitions [1], is the task of extracting the right answer from text snippets identified in large collections of documents where the answer to a natural language question lies.

* Corresponding author. Tel.: +34-965903772; fax+34-965909326
Email address: rafamt@dlsi.ua.es (Rafael M. Terol).

[1] The Text REtrieval Conference(TREC) is a series of workshops organized by the National Institute of Standards and Technology (NIST), designed to advance the background in Information Retrieval (IR) and QA

Open-domain textual QA systems are defined as capable tools to extract concrete answers to very precise needs of information in document collections. For instance, in open domains, a system can respond to society questions such as *where was Marilyn Monroe born?*, *what is the name of Elizabeth Taylor's fourth husband?*; geography questions such as *where is Halifax located?* and so on. Examples of these kinds of QA systems in open domains can be located in authors such as Moldovan [12], Sasaki [19], Vicedo [20], Zukerman [21], and so on. These types of QA systems locally process document collections discarding the access to internet information sources.

In restricted domains, Frequently Asked Question (FAQ) systems are often used to obtain common answers to a restricted set of questions that users need. These FAQ systems handle a database where the list of questions and their related answers are stored. Thus, the FAQ system allows users to choose one of the possible questions that the system is able to answer by way of searching in the database for the answers related with that question. The natural language questions do not consider by these FAQ systems and, the increment in the questions treated by the system require the user to compare if the question is matched with the large number of the answered questions. For these reasons, these FAQ systems are replaced by QA systems over restricted domains. Nowadays, textual QA is also exhibited in restricted domains such as clinical [4], tourism [1], medical [16] and so on. These system are described in the next background section.

According to official results of the QA track at the last TREC conference, QA systems in open domains are between 30% and 40% of precision [2] . In a restricted domain such as medical domain, it is necessary to highly improve this score due to the critical information that is handled in these medical areas where erroneous information can originate serious risks to people's health (no answer is better than incorrect answers).

This is the reason why our research effort is directed towards the textual QA on medical domain retrieving the information from internet websites. There exists a lot of feasible medical information towards internet, the largest network in the world. This fact increases the importance of evaluating the quality of information on medical websites because anyone can create a website and can put any medical information on this website. This medical information would not be accurate or true.

In this paper, a QA system is presented. This QA system is capable of working over any restricted domain. The adaptation to the system medical domain (medical QA system) is also exhibited. The medical QA system is able to answer medical questions according to a generic question taxonomy. In the following sections, the main features of the QA system are described focusing in detail the question analysis

[2] This evaluation measure gives the accuracy of the QA system

performance. Section 2 introduces the state of the art of QA systems. In Section 3, we show the motivation of working in QA over medical domain. Section 4 details the modulate architecture of the restricted domain QA system and its adaptation to the medical domain. In Section 5, we describe the evaluation task and show the obtained results by our medical QA system. Section 6 discusses the contribution of our research work. The last section summarizes the present article.

2 Background

QA performance requires complex natural language processing (NLP) techniques. The core of our QA system is the text processing by way of logic forms. In the following sections, this complex NLP technique is defined. A logic form is a way of representing natural language sentences. Other authors employ logic forms in their QA systems. Concretely, Dan Moldovan [12] developed an open domain QA system, and Diego Mollá [14] designed an open domain QA system capable of answering natural language questions in the frame of the commands of the UNIX operating system. In Moldovan's QA system, the identification of the predicates is based on the format of Logic Form Transformation of eXtended WordNet [6] while Mollá identifies the predicates using a more complex terminology based on logic treatment. In order to focus their QA systems on open domains, Moldovan and Mollá employ complex inference rules in the logic forms treatment performance.

Moreover, the use of these open-domain textual QA systems in restricted domains such as medical domain do not produce good results because these systems use natural language processing generic resources such as WordNet [3] [10] which is not specialized in medical terminology. When QA systems are directed to restricted domains, it is necessary to acquire rich knowledge resources of the domain that allows the system to understand the meaning of the treated information in the user's question and documents. Chung et al [2] presented a practical QA system in the meteorology domain that extracts information about the weather every hour from the website of the Korea Meteorological Administration. This information is structured and locally stored in a database management system (DBMS). The knowledge is obtained by consulting a domain-dependent ontology for recognizing weather events, and the domain independent ontology for place names. Rinaldi et al [18] shows the adaptation to the genomics domain of an existing QA system. The knowledge was extracted from several resources such as UMLS [8], SWISS-PROT, OMIM, GeneOntology, GenBank and LocusLink. As an adaptation of the ExtrAns system [14] to the new genomics domain, this system uses the minimal logical forms to perform the semantic representation of documents and questions. Niu & Hirst previous work [15] showed that current technologies for factoid QA [4]

[3] WordNet is a large lexical database of the English language

[4] A factoid question is a fact-based, short answer question such as When did Lennon die?

in open domains were not adequate for clinical questions, whose answers must often be obtained by synthesizing relevant context. To adapt to this new characteristic of QA in the medical domain, they exploited the relations between the semantic classes in medical text.

As shown in the present section, different ways of processing logic forms are applied in the open-domain QA performance. Also, open-domain QA systems can be adapted to restricted domains. In the following sections, our QA system based on the processing of logic forms is presented. The features that allow the portability of the QA system to a new domain (the medical domain) are also presented. These portability features imply that our QA system runs as a medical QA system.

3 Motivation

There exists several agents that can interact in the clinical domains such as doctors, patients, laboratories and so on. All of them need quick and easy ways to access electronic information. Access to the latest medical information helps doctors to select better diagnoses, helps patients to know about their conditions, and allows to establish the most effective treatment. These facts produce a lot of information and different types of information between these agents that must be electronically processed. For example, people want to find competent medical answers to medical questions: when they have some unknown symptoms and want to know what they could be related to, or when they want to know another medical opinion about the best way to treat their disease, or when they can ask experienced doctors any medical questions related to any unknown symptoms or their state. All these features conclude that the number and the type of medical questions that a medical QA system can respond to is very great.

These reasons motivated us to adapt the QA system to the medical domain. This medical QA system is capable of answering medical questions according to a medical question taxonomy. This question taxonomy is based on the study developed by Ely *et al* [5] whose main objective is to develop a taxonomy of doctor's questions about patient care that could be used to help answer such questions. In this study, the participants were 103 Iowa family doctors and 49 Oregon primary care doctors. The authors concluded that clinical questions in primary care can be categorized into a limited number of generic types. A moderate degree of interrater reliability was achieved with the the taxonomy developed in this study. The taxonomy may enhance the understanding of doctors' information needs and improve the ability to meet those needs. According to this question taxonomy, the ten most frequent questions formulated by doctors are ranked in the following enumeration:

(1) What is the drug of choice for condition x?
(2) What is the cause of symptom x?

(3) What test is indicated in situation x?
(4) What is the dose of drug x?
(5) How should I treat condition x (not limited to drug treatment)?
(6) How should I manage condition x (not specifying diagnostic or therapeutic)?
(7) What is the cause of physical finding x?
(8) What is the cause of test finding x?
(9) Can drug x cause (adverse) finding y?
(10) Could this patient have condition x?

Thus, our medical QA system must be able to answer natural language questions according to this set of ten generic medical questions, discarding other questions (medical and from other domains). The fact that our QA system is only able to answer questions in this question taxonomy produces on one hand a lower recall but on the other hand a higher precision with the aim that our system will be very useful in the medical domain according to this question taxonomy.

This adapted domain QA system (in this case, medical domain) uses complex NLP techniques as logic forms treatment. The main differences in the logic forms of our QA system and those of Moldovan and Mollá are based on the method of derivation of the logic forms, the method of identifying the predicates in the logic forms and the complexity of the inference rules in the treatment of the logic forms. On the one hand, the QA systems of Moldovan and Mollá derive the logic forms through the syntactic analysis of the sentence while, on the other hand, our QA system derives the logic form through the dependency relationships between the words. As Courtin & Genthial [3] said, the processing based on syntactic analysis allows to add some semantic information on words. In open-domains, this method of derivation of the logic forms improves the knowledge of the system. On the other hand, in restricted domains where there exists other knowledge resources, the derivation of the logic form through the dependency relationships between the words is more concise. Also, in our QA system as in Moldovan's QA system, the identification of the predicates is based on the format of Logic Form Transformation of eXtended WordNet. In order to focus our QA system in restricted domains, in the logic forms treatment task, our inference rules are deeper than the inference rules applied by Moldovan and Mollá.

The next section details the modulate architecture of our QA system capable of answering the questions formulated according to a question taxonomy. Concretely, we show the adaptation to the specific medical domain taxonomy, implemented by means of the medical QA system.

4 QA system architecture

The main components (modules) of our QA system could be summarized in the following steps:

(1) Question Analysis.
(2) Document Retrieval.
(3) Relevant Passages Selection.
(4) Answer Extraction.

These components are related to each other and process the textual information available on different levels until the QA process has been completed.

The natural language questions formulated to the system are processed initially by the question analysis component. This process is very important since the quantity and quality of the information extracted in this analysis will condition the performance of the remaining components and therefore, the final result of the system.

A part of the information obtained from this question analysis process is used by the document retrieval module to perform a first selection of documents from websites. In a restricted domain the document collections are frequently updated and this fact derives high maintenance costs of updated document collections locally stored. This is the main reason why this task is remotely performed using the google search service. The obtained result is a very reduced subset of the documentary database in the websites.

Subsequently, the relevant passages selection module performs a more detailed analysis of the relevant documents subset with the objective of detecting those reduced text fragments that are susceptible of containing the search answer.

Finally, the answer extraction module processes the small text fragments set obtained from the previous process with the purpose of locating and extracting the search answer. Figure 1 graphically shows the execution sequence of these processes and the relationships to each other modules.

The computational cost of this complex process is primarily dependant on two main factors: the speed of the internet connection in the tasks of document retrieval and named entities recognition, and the logic form derivation task. The temporal costs derived from the speed of the internet connection would be lower if the document collection and the knowledge resources (presented in the following subsections) were locally stored because our system is also able to locally work with these resources. We prefer to remotely work with these resources because they are frequently updated (new drugs, new releases of knowledge resources, and so on). Moreover, with the aim of running this medical Q-A system in the most common operating systems, the JAVA™ platform has been used in the development phase.

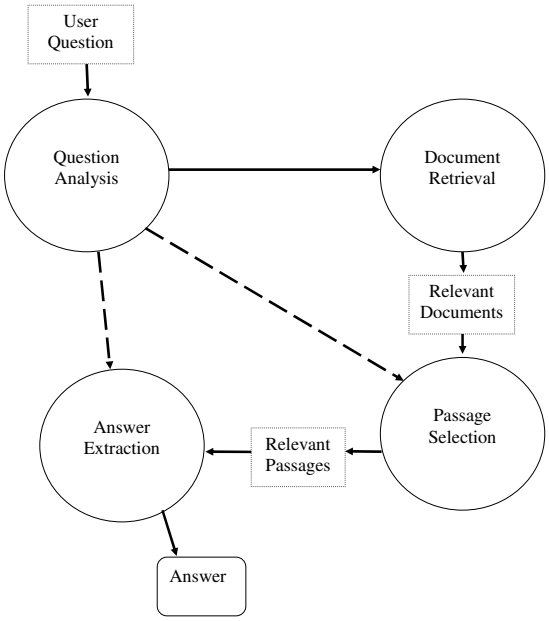

Fig. 1. Medical QA System Modulate Architecture

The needs of persistent information are stored in the file system of the operating system. Thus, the dependencies between the database management systems and the operating systems are avoided. Considering these development features of accessing the resources via internet, the medium temporal cost of answering a question using the QA system is around 8 seconds.

The subsection 3.1 presents how the QA system performs a previous preprocessing of the sentences (questions and possible answers). The subsection 3.2 shows the portability features that allow the QA system to run as a medical QA system. Then, the rest of the subsections (from 3.3 to 3.6) describe the main components of the medical QA system architecture.

4.1 Preprocessing of the sentences

This previous preprocessing of the sentences allows the main modules to infer logic forms of sentences and obtain similarity relationships between verbs in the WordNet [10] lexical database.

4.1.1 Inferring Logic Forms of Sentences

Our medical QA system makes use of the logic forms of the sentences with the aim of simplifying the sentence treatment process. The logic form of a sentence is derived through applying NLP rules to the dependency relationship of the words in the sentence.

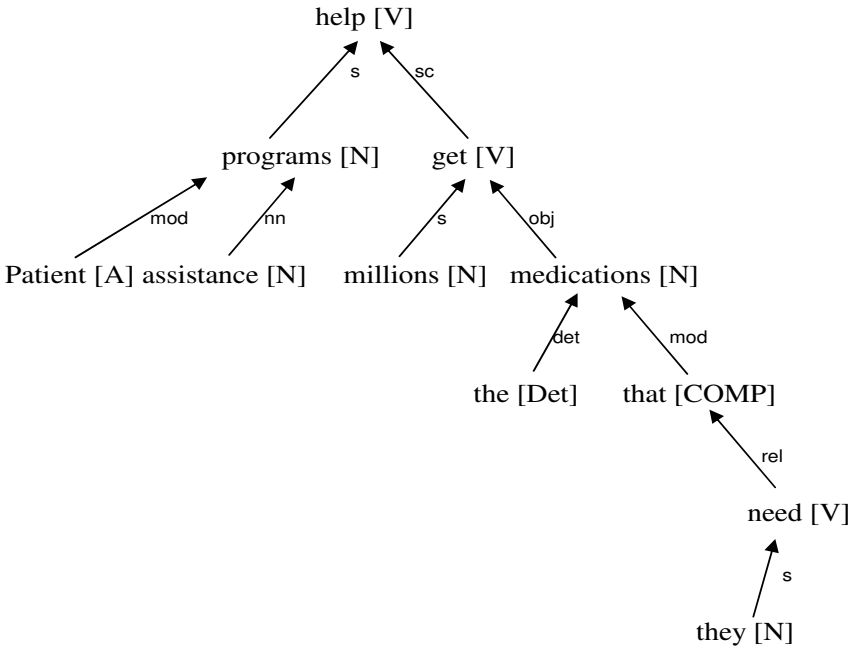

Fig. 2. Dependency tree of the sentence

4.1.1.1 Getting Dependency Relationships The first step necessary to infer the logic form of a sentence is to obtain the dependency relationships between the words of the sentence. The NLP resource used to obtain the dependency relationships between the words of the sentence is MINIPAR [7], a broad-coverage parser.

According to the definition proposed by Lin [7], a dependency relationship is an asymmetric binary relationship between a word called head (or governor, parent), and another word called modifier (or dependent, daughter). Normally the dependency relationships constitute a tree that links all the words in the sentence. This dependency tree has different levels of words because a word in the sentence may have different modifiers, but each word may modify at most one word. The root of the dependency tree does not modify any word. It is also called the head of the sentence.

For example, Figure 2 shows the dependence tree of the sentence "*Patient assistance programs help millions get the medications that they need*". The lexical category of each word is shown inside the brackets behind the word. These lexical categories can be noun (N), verb (V), adjective (A), and so on. Each one of the arrows label the dependency relationship between the modifier and the head. These dependency relationships can be s (subject), mod (modifier), obj (object), and so on. In this example, the verb 'to help' is the head of the sentence (the root of the dependency relationship).

4.1.1.2 Logic Form Derivation Once the dependency relationships have been acquired, the next step to automatically infer the logic form of the sentence is

the analysis of these dependency relationships between the words of the sentence. Then, the logic form derivation is a compositional process that starts in the leaves of the dependency tree, continues through the ramifications of the dependency tree and ends in the root of the derivation tree. Thus, the logic form is inferred on the one hand by the application of simple NLP rules to the leaves of the dependency tree and, on the other hand, by the application of complex NLP rules to all the pairs (modifier, head) in the dependency tree. This distinction between single and complex NLP rules is produced because in the leaves of the dependency tree there does not exist any dependency relationship in which the word is the head of the dependency relationship while in the ramifications and in the root of the dependency tree dependency relationships do exist.

To design the single NLP rules only the lexical category of the word has been contemplated while in the design of the complex NLP rules the lexical category of the head, the lexical category of the modifier, the type of dependency relationship and the relative position of the modifier (before the head or after the head) have been considered. Table 1 shows some simple NLP rules and Table 2 describes some complex NLP rules. In these tables, the Leaf column expresses the lexical category of the leaf, the LCH column describes the lexical category of the head in the dependency relationship, the LCM column expreses the lexical category of the modifier in the dependency relationship, the DR column shows the type of dependency relationship, the MP column expresses the relative position of the modifier with respect to the head, and the LF column shows the inferred logic form.

Table 1
Subset of simple NLP rules applied to the leafs in the dependency tree

Leaf	**LF**
Det	void
A	lemma*:JJ(*new x var*)*
N	lemma*:NN(*new x var*)*

Table 2
Subset of complex NLP rules applied to dependency relationships

LCH	**LCM**	**DR**	**MP**	**LF**
N	*Det*	*det*	*before*	lemma of head*:NN(*new x var*)*
A	*A*	*mod*	*before*	modifier LF + lemma of head*:JJ(*modifier x var*)*
VBE	*N*	*subj*	*before*	modifier LF + lemma of head*:VB(*new e var, modifier x var, new x var*)*
VBE	*A*	*pred*	*after*	head LF + *Atributo:IN(*head e var, modifier x var*)* + modifier LF

The assignation of predicates and arguments to the lemma of the words is based on

the codification applied by Logic Form Transformation of eXtended WordNet [6], a lexical resource based on logic forms. This codification depends on the part-of-speech (lexical category) of the words:

- Noun: An x-type argument is assigned to the predicate of this word. This argument uniquely identifies this predicate in the logic form. For instance, the noun "house" could be codified by the predicate "house:NN(x1)".
- Verb: An e-type and two x-type arguments are assigned to the predicate of this word. The first one uniquely identifies this predicate (the action of the verb) in the logic form and the other ones denote the subject and the object of the word. If the verb is intransitive then the object argument must be dummy. As an example, the noun "take" could be codified by the predicate "take:VB(e1, x1, x2)".
- Adjective: An x-type argument is assigned to the predicate of this word. This argument uniquely recognizes this predicate (the property denoted by the adjective) in the logic form. For instance, the adjective "young" could be codified by the predicate "young:NN(x1)". When the adjective modifies a noun (there exists a dependency relationship from the adjective to the noun) then both predicates in the logic form are instantiated by the same x-type argument. For instance "young man" could be codified as "young:JJ(x1) man:NN(x1)".
- Adverb: An e-type argument is assigned to the predicate of this word. This argument uniquely identifies this predicate (the action expressed by the adverb) in the logic form. As an example the adverb "clearly" could be codified by the predicate "clearly:RB(e1)".
- Preposition: A combination between the x-type and e-type arguments can be assigned as the two arguments of this predicate that only link the dependency relationship between two other predicates. For instance, the expression "south of America" could be codified as "south:NN(x1) of:IN(x1, x2) America:NN(x2)" while the expression "go to the airport" could be codified as "go:VB(e1, x1, x2) to:IN(e1, x3) airport:NN(x3)".

We summarize this complex process of inferring the logic form of a sentence through the following example in the sentence "The aspirin is effective". The first step is to find the dependency relationships between the words in the sentence. Figure 3 shows the dependency tree. The second step consists of applying the simple NLP rules to the leaves of this dependency relationship and obtaining the predicates of the logic form derived in these leaves (see Table 3). The next step is based on applying the complex NLP rules to the ramifications and the root of the dependency tree deriving the logic form (see Table 4).

Table 3
Simple NLP rules applied to the leafs in the dependency tree

Leaf	**LF**
The [Det]	void
effective [A]	*effective:JJ(x1)*

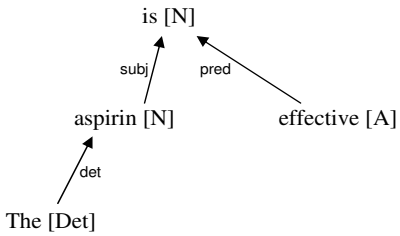

Fig. 3. Dependency tree of the sentence

Table 4
Complex NLP rules applied to dependency relationships

LCH	LCM	DR	MP	LF
aspirin [N]	*The [Det]*	*det*	*before*	*aspirin:NN(x2)*
is [VBE]	*aspirin [N]*	*subj*	*before*	*aspirin:NN(x2) be:VB(e1, x2, x3)*
is [VBE]	*effective [A]*	*pred*	*after*	*aspirin:NN(x2) be:VB(e1, x2, x3)*
				Atributo:IN(e1, x1) effective:JJ(x1)

Once all these rules have been applied to the dependency tree of the sentence "The aspirin is effective", the logic form is inferred as "aspirin:NN(x2) be:VB(e1, x2, x3) Atributo:IN(e1, x1) effective:JJ(x1)". Note that the verb "to be" is intransitive. This fact produces in the logic form that on the one hand the argument of its predicate that represents the object (x3) is dummy and, on the other hand, the predicate "Atributo" links the dependency relationship between the verb and the adjective.

Our NLP technique used to infer the logic is different to other techniques that accomplish the same goal such as Moldovan's [11] that takes as input the parse-tree of a sentence, or Mollá's [14] that introduces the flat form as an intermediate step between the sentence and the logic form.

This generic NLP resource based on inferring the logic forms of the sentences is used by our medical QA system in the performance of question analysis (deriving the logic forms of the questions) and answer extraction (deriving the logic forms of the sentences that would contain the answer).

4.1.2 Similarity Relationships between Verbs

In spite of the fact that UMLS is a rich resource in medical expressions, it does not contain much information related to verbs because the verbs should not be domain-independent. For this reason our system uses WordNet [10] to extract the similarity relationships of one verb to another. WordNet is a database of word meanings and lexical relationships that records the semantic relations between the synonym sets, also called synsets. A synset can be defined as a group of synonym words. These synsets are related to each other according to different relations: synonymy, hyponymy, hyperonymy, coordinate terms, holonymy, meronymy, antonymy, and so

on.

4.2 Portability of the system to the medical domain

To adapt the QA system to the medical domain it is necessary to obtain medical knowledge by way of medical named entities recognition, and develop the patterns associated to each one of the treated generic medical questions.

4.2.1 Medical Named Entities Recognition

Our medical QA system needs to recognize the medical entities in the sentences focusing on the processing in the different phases of the QA process. The medical named entities recognition performance is developed by using the Unified Medical Language System (UMLS) [8], a resource of the language of biomedicine and health. A great number of concepts, relationships and definitions contained in UMLS have been derived from the Medical Subject Headings (MeSH) vocabulary [5] . Concretely, our system uses the UMLS knowledge source called Metathesaurus [9] to accomplish this goal. The UMLS Metathesaurus contains information about biomedical and health related concepts (meanings) facilitating mapping free-text entries to biomedical and health terminologies. The UMLS Metathesaurus is organized by concept. These concepts have assigned, at least, one semantic type (category). Table 5 shows an example of the mapping free-text entries to biomedical and health terminologies by way of concepts and semantic types in the UMLS Metathesaurus. On the one hand, the CUI column uniquely identifies the concept while the CN column shows the name of the concept, and on the other hand, the TUI column uniquely identifies the semantic type while the STY column describes the name of the semantic type associated to the concept. Thus, our medical named entities recognition module is based on dictionary. This module retrieves from the UMLS Metathesaurus all the information relative to the concept and the semantic types of the free-text received as argument. The retrieval of this information from the UMLS Metathesaurus is performed by consuming the UMLS Metathesaurus webservice through Simple Object Access Protocol (SOAP), an XML-based messaging protocol. The processing of this retrieved information is locally performed.

Even though our QA system is able to locally work with the UMLS Metathesaurus, this feature is actually discarded because this resource is frequently updated with new releases. The fact that the execution time decreases in a few seconds by locally working with this resource would suppose the following disadvantages: to detect when a new release has been published, to download this new release, to replace

[5] MeSH is a huge controlled vocabulary created by the United States National Library for the purpose of indexing journal articles and books in the life sciences.

Table 5
Free-text entries mapped to UMLS

Free-text	Concept Info.		Semantic Type Info.	
	CUI	CN	TUI	STY
acetylsalicylic acid	C0004057	*Aspirin*	*T109*	*Organic Chemical*
			T121	*Pharmacologic Substance*
high blood pressure	C0020538	*Hypertension*	*T047*	*Disease or Syndrome*
cephalgia	C0018681	*Headache*	*T184*	*Sign or Symptom*

the previous installation with the new release, and to make possible changes in the software that interacts with the new release.

4.2.2 Pattern generation

This off-line task consists of the definition of the patterns that identify each generic question. These patterns are composed by a combination of types of medical entities and verbs. These patterns can be generated according to two different methods: the first one consists of the easy process of definition of patterns by an advanced user of the system, and the second one consists of the automatic generation of the patterns through the processing of questions according to the question taxonomy. We are going to describe these two different ways of generating patterns:

Manual Pattern Generation. The manual definition of these patterns is presented in Figure 4. The advanced user of the system has to identify the types of medical entities and verbs that must match in the generic question. The automatic expansion of these verbs according to their similarity relationships with other ones in the WordNet lexical database is also performed. The following step consists of setting the medical entities lower threshold (MELT) and the medical entities upper threshold (MEUT) of each pattern. On the one hand, MELT can be defined as the minimum number of medical entities that must match between the pattern and the question formulated by the user and, on the other hand, MEUT can be defined as the maximum number of medical entities that can match between the pattern and the question formulated by the user. Finally, the last step consists of the manual setting of the possible expected answer types.

Supervised Automatic Pattern Generation. The automatic generation of these patterns by the system is performed through the processing of questions matched to the question taxonomy as shown in Figure 5. Thus, the first step consists of the derivation of the logic form associated to each question. The next step is the Medical Named Entities Recognition in the logic form of those predicates whose type is noun (NN) or complex nominal (NNC) including their possible adjective modifiers (JJ). The third step is the recognition of the main verb in the logic form and the

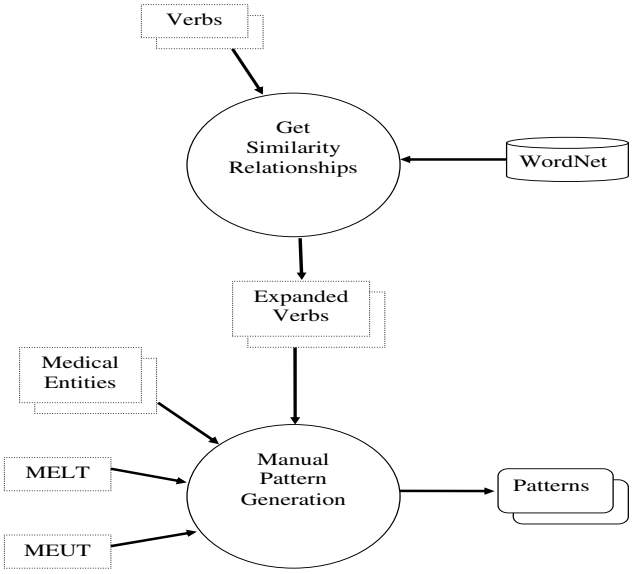

Fig. 4. Manual Pattern Generation Task

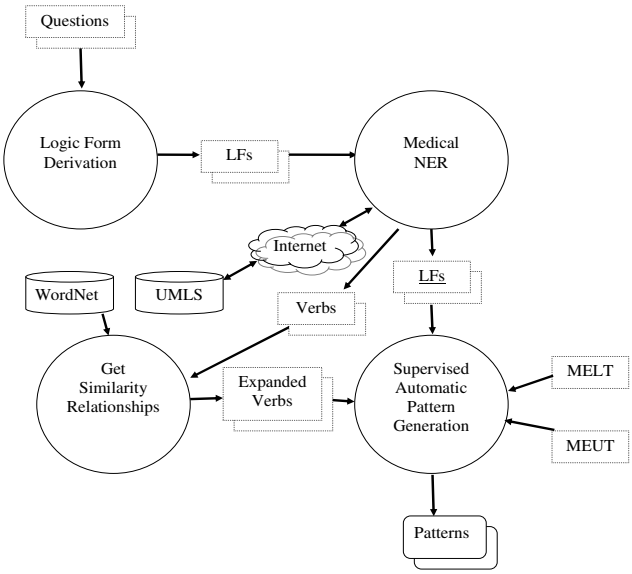

Fig. 5. Supervised Automatic Pattern Generation Task

automatic expansion of this main verb in the logic form through the similarity relations with other verbs in the WordNet lexical database. The next step consists of the automatic setting of the MELT whose score is set to the number of medical entities in the logic form minus one, and the automatic setting of the MEUT of which the score is set to the number of medical entities in the logic form. Finally, the last step consists of the manual setting of the possible expected answer types. This task is supervised by an advanced user of the system that can modify the results obtained by the system in each step.

4.3 Question Analysis

The Question Analysis performance consists of classifying and analyzing the natural language questions that users can ask. This computational process is based on two different tasks:

- **Question Classification:** assigning one of the generic patterns to each one of the questions that the user asks our system.
- **Question Analysis** performing a complex process on the question according to the matched pattern and its respective matched generic question.

4.3.1 Question Classification

This Question Classification task starts after the user enters the question into the system. In this implementation of the QA adapted to the medical domain, ten classes of user questions are managed according to the ten generic questions treated by the system. Then, this task has to decide if the user question belongs to one class (matches with one of the generic questions) or not. To accomplish this goal, this task focuses on the treatment of question forms derived from the user questions according to the steps shown in the Figure 6. Thus, the first step consists of inferring the logic form of the question entered to the system. The second step is the extraction of the main verb in the logic form. The next step is the recognition of the medical entities of those predicates whose type is noun (NN) or complex nominal (NNC) including their possible adjective modifiers (JJ). The fourth step is the analysis of the question form setting the medical entities score in question (MESQ). MESQ can be defined as the number of medical entities in the logic form of the question. The next step consists of finding those patterns of questions of which the list of verbs contains the main verb of the logic form and $MELT \leq MESQ \leq MEUT$. The next step consists of setting the entities matching measure (EMM) which is defined as the number of medical entities that match between the question and the pattern. Finally, the last step is the selection of the pattern whose difference between EMM and MELT is the lowest one.

4.3.2 Question Analysis

Once the user question is matched to a generic question pattern from one of the ten generic questions treated by the system, this Question Analysis task firstly captures the semantics of the user question. As mentioned before, WordNet and UMLS Metathesaurus are used in this performance. The following step consists of the recognition of the expected answer type. These medical answer types can be diseases, symptoms, dose of drugs, and so on, according to the possible answers to the ten generic questions treated by the system. After that, the keywords are identified. These question keywords are directly recognized by applying a set of heuristics

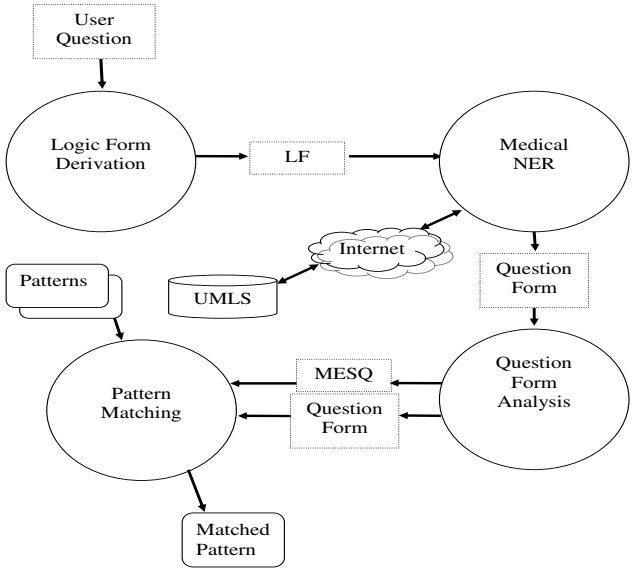

Fig. 6. Question Classification Task

to the predicates and the relationships between predicates in the logic form. Like question keywords our QA system identifies complex nominals and nouns recognized as medical expressions (using Medical Named Entities Recognition) including their possible adjective modifiers, the rest of the complex nominals and nouns including their possible adjective modifiers and the main verb in the logic form. For instance, in the part of the logic form "... high:JJ(x3) blood:NN(x1) NNC(x3, x1, x2) pressure:NN(x2) ...", the predicate *x3* is recognized as a *Disease or Syndrome* and then "high blood pressure" is treated as a keyword. These question keywords can be expanded by applying a set of heuristics. For example, medical expressions can be expanded using similarity relations given by UMLS Metathesaurus. Thus, according to UMLS Metathesaurus, "high blood pressure" can be expanded to "hypertension".

This set of question keywords is sorted by priority, so if too many keywords are extracted from the question, only a maximum number of keywords are searched in the information retrieval process.

4.4 Document Retrieval

Even though the document retrieval module can retrieve locally stored documents, its remote facility retrieves the relevant documents from medical websites using the google search service. These medical websites can be sorted from the previously defined medical website classification. This medical website classification is performed before the real-time execution of the google search engine and consists of defining the different medical website classes where our system can retrieve the medical documents. Once these medical website classes have been defined, an addi-

tional task that consists of relating the generic questions and these medical website classes can be defined but it is not necessary. Note that a medical website class can be related to more than one generic question, and a generic question can be associated to more than one medical website class. Thus, this association relates each one of the generic questions and the medical websites that can answer them.

Then, this document retrieval engine can start retrieving those relevant documents from medical websites whether there exists or not the association between the searched generic question and the medical websites.

4.4.1 Document Retrieval by way of Medical Websites Classes

When the treated generic question has been related to at least one medical websites class then the google search engine retrieves the relevant documents according to the question keywords in these medical websites.

4.4.2 Document Retrieval by way of MFC Algorithm

When the treated generic question has not been related to any medical website class then we apply our most frequent classes (MFC) algorithm. This algorithm calculates the most frequent medical website classes that rightly answer the treated generic question in the latest searches. Thus, the google search engine retrieves the relevant documents according to the question keywords in the medical websites that belong to these most frequent medical website classes. The update of the MFC for the treated generic question is produced using an adaptation of the LRU algorithm for database disk buffering [17]. This task consist of updating the MFC for the treated question with the actual medical website classes where the right answer can be found.

4.5 Relevant Passage Selection

Once the medical documents are retrieved, this Relevant Passage Selection process consists of extracting the sentences in these medical documents that could answer the user question. These sentences are extracted by applying a technique based on comparing the question keywords in the documents and, those sentences that at least contain a question keyword are extracted from the document and are evaluated by the next Answer Extraction module that decides if the sentence rightly answers the user question.

4.6 Answer Extraction

This module extracts the answer by analyzing the sentences extracted by the previous relevant passage selection module. This process is performed by applying the following steps to each one of the retrieved sentences: the first one consists of inferring the logic form of the sentence and identifying the main verb in this logic form; the following step is to verify if this main verb belongs to the set of verbs that can answer the generic question; the third step is the recognition of the medical entities in the logic form; the next step consists of comparing if the medical entities searched as the answer is found in the logic form; and finally, the last step is the analysis of the predicates that relate the candidate answer, the main verb and the rest of the medical entities in the logic form (answer form). This process produces an Answer Ranking. In a valid answer, the verb can uniquely relate two medical entities considering this feature as a direct link. Also, IN-type predicates can take part in the relation between the two medical entities considering this feature as a connect link. Our system differently scores these two links: 1 for the direct link, and 0.8 for the connect link. To rank the answer, our system applies the link measure defined as:

$$LM = \frac{\sum link_i}{\# \, links}$$

For example, if a user formulates the system with the question “Which drugs are associated with the high blood pressure problem?”, this question is classified according to the first generic question “What is the drug of choice for condition x?”. Continuing with the processing, the answer extraction module receives as input the following sentences: “Cozaar treats hypertension” and “Hyzaar is indicated in the management of hypertension”. The logic form associated to the first sentence is “cozaar:NN(x1) treat:VB(e1, x1, x2) hypertension:NN(x2)” while the logic form associated to the second sentence is defined as “hyzaar:NN(x1) indicate:VB(e1, x1, x4) in:IN(e1, x3) management:NN(x3) of:IN(x3, x2) hypertension:NN(x2)”. The answer form associated to the first logic form is instantiated as “Pharmacologic_Substance:NN(x1) treat:VB(e1, x1, x2) Disease_or_Syndrome:NN(x2)”. Only a direct link (the *treat* verb) relates both medical entities (Pharmacologic_Substance and Disease_or_Syndrome). In this case LM=1. The answer form associated to the second logic form is instantiated as “Pharmacologic_Substance:NN(x1) indicate:VB(e1, x1, x4) in:IN(e1, x3) management:NN(x3) of:IN(x3, x2) Disease_or_Syndrome:NN(x2)”. A direct link (the *indicate* verb) and two connect links (*in* and *of*) relate both medical entities (Pharmacologic_Substance and Disease_or_Syndrome). In this case LM=0.8. Then, the answer ranking according to the LM scores would be: Cozaar and Hyzaar. These two answers would be the results returned by the system. LM ranks the answers according to the length of the paths between the treated medical entities. Thus, short paths would be in header positions relative to long paths.

5 Results

The evaluation of the medical QA system is based on the question analysis module, the core of the system, because the good performance of its question classification task (rightly classifying the formulated question into one of the generic questions) finally derives in the increasing of the precision of the system. Despite the fact that open-domain QA systems can be evaluated according to TREC and CLEF [6] evaluation tracks, when a QA system is directed to any restricted domain do not exist these kinds of evaluation tracks. This is the main motivation why the evaluation of the question classification task is based on the evaluation presented by Chung *et al.* in their previous research work [2]. Thus, a pilot evaluation task applied to the evaluation of the question classification performance has been developed involving a group of people that did not work on the design and development phases of the QA system. These people received several instructions about the manual construction of these types of questions to manually create fifty questions according to the ten generic questions answered by the system (GQ_1: five questions that are matched to the first generic question; ...; GQ_{10}: five questions that are adjusted to the tenth generic question.). Also, the OQ question set that is composed of 200 questions of the last QA English Track at CLEF 2005 conference is also included to evaluate the robustness of the question classification task in a noisy environment.

Figure 7 shows how the question classifier task is able to classify each one of the given questions in one of the following classes of questions:

- **GE**: This class of questions include each one of the ten generic questions. Thus, GE_1 corresponds with the generic question "What is the drug of choice for condition x?", GE_2 is matched with the generic question "What is the cause of symptom x?",..., and GE_{10} is arranged with the generic question "Could this patient have condition x?".
- **OE**: The rest of the questions from other domains.

Then the evaluation task consist of checking if each one of the 250 evaluation questions (GQ_1, ..., GQ_{10} and OQ) have been correctly classified in the appropriate class of questions (GE_1, ..., GE_{10} or OE). As an evaluation measure, we apply the precision measure (P) defined as $P = \frac{\#\ correctly\ classified\ questions}{\#\ classified\ questions}$.

In order to show the results obtained in this question classification task, Table 6 shows the obtained results in the evaluation of each subset of evaluation questions while Table 7 presents these summarized results according to the generic set of evaluation questions. The Class column expresses the class of questions that we are evaluating. The Related Class column shows the correct related class associated to each classified class. The Questions column presents the number of classified

[6] Similar to TREC, CLEF is other system evaluation campaign where QA systems can be tested, tuned and evaluated.

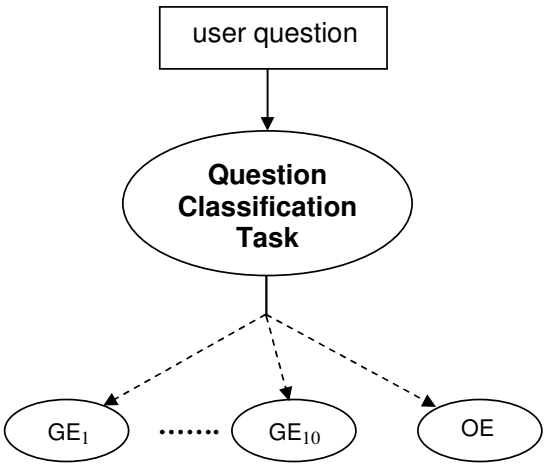

Fig. 7. Question Classification Task

questions. The number of 5 questions per class [7] and 200 noisy questions has been empirically established in the pilot evaluation task but this fact does not mean that the classifier is only able to classify this number of questions. The classifier, as the rest of components of the QA system, does not consider this number of questions to perform their functions. So, the QA system is able to sequentially manage an unlimited number of questions. The Correct column indicates the number of questions that have been correctly classified according to the related class. The Precision column shows the precision of this classification task that agrees with the presented evaluation measure.

Table 6
Detailed Evaluation of the Question Classification Task

Classified Class	Related Class	Questions	Correct	Precision
GQ_1	GE_1	5	5	1
GQ_2	GE_2	5	5	1
GQ_3	GE_3	5	3	0.6
GQ_4	GE_4	5	4	0.8
GQ_5	GE_5	5	5	1
GQ_6	GE_6	5	4	0.8
GQ_7	GE_7	5	4	0.8
GQ_8	GE_8	5	3	0.6
GQ_9	GE_9	5	5	1
GQ_{10}	GE_{10}	5	4	0.8
OQ	OE	200	194	0.97

[7] 5 questions per class according to the question taxonomy

Table 7
Summarized Evaluation of the Question Classification Task

Classified Class	Related Class	Questions	Correct	Precision
GQ	GE	50	42	0.84
OQ	OE	200	194	0.97
$Overall$	–	250	231	0.944

According to the overall row in Table 7, the precision score of the question classifier task is 94,4%. This good score will positively condition the right performance of the following parts of this QA process in the medical domain.

6 Discussion

It is well known that there exists a lot of information needs related to the different medical areas and specialities. Most of the on-line information in the health and medical areas are unknown to people outside of these areas including health care professionals. These information needs can be solved by applying the medical QA system capable of answering medical questions by retrieving the information from medical websites discarding any other wrong medical information that anybody can put on different websites. According to the proposed architecture the medical QA system can be easily transformed to a client-server application on the web accessed through a web-browser. Thus, the use of the medical QA system would be accessible to everybody.

The main novelty of the medical QA system is that the information can be retrieved from internet websites in comparison to most QA systems (in open and restricted domains) that only retrieve locally stored information in a known host. In spite of the medical question taxonomy presented in this article, the extension to other medical questions can be easily performed. Due to the efficient resources and techniques used by the medical QA system, the average temporal costs are round about eight seconds per answered question.

Also, with the aim to improve the temporal costs in answering the medical questions, each treated medical question can be searched in the medical websites considered by the system administrator. If this feature is not considered then the system automatically applies an adaptation to our task of the LRU algorithm used by the operating systems and the database management systems in the memory management performance. This algorithm considers the medical websites where the system retrieved the documents that rightly responded to this class of question in previous executions of the system, and orders them according to the number of right responses retrieved in each medical website.

The software engineering rules that treat the module coupling and module cohesion properties in an object-oriented context have been applied in the design of the medical QA system architecture. For this reason the medical QA system can also be easily extended to other domains. This fact only implies the adaptation to the new domain of the system's submodule that performs the entities recognition task, and the indications of which are the right websites dependant on the new domain that contain the information in which the answers can be extracted.

7 Summary

QA is applied to medical disciplines in modern QA over restricted domains. It allows users to efficiently obtain a list of answers to medical questions. The medical QA system presented in the present article is able to answer these questions according to a medical question taxonomy. Thus, the medical QA system offers tools to automatically define the functional patterns of a new medical question towards a set of matched questions to this new medical question. Once these functional patterns have been automatically created, the new medical question is able to be answered by the medical QA system, in conjunction with the rest of these generic medical questions. Also, the medical websites where the system can find the right answers to the new question can be given easily as an input of the system. This guide to medical websites will improve the temporal costs of the system in answering this class of medical questions. The core of the medical QA system is the logic form treatment. This complex process is produced by applying advanced NLP techniques. The logic form of a sentence is derived through applying NLP rules to the dependency relationship of the words in the sentence. The NLP resource used to obtain these dependency relationships is MINIPAR [7], a broad coverage parser. Other NLP resources are used in this complex process: on the one hand the WordNet lexical database [10] is used to extract the similarity relationships between the verbs and, on the other hand, the UMLS Metathesaurus [9] is used to recognize the medical named entities in the text. In spite of the fact that this QA system has been adapted to the medical domain, it also can be adapted to other restricted domains.

References

[1] Farah Benamara. Cooperative Question Answering in Restricted Domains: the WEBCOOP Experiment. In *ACL 2004 Workshop on Question Answering in Restricted Domains*, Barcelona, Spain, July 2004.

[2] Hoojung Chung, Young-In Song, Kyoung-Soo Han, Do-Sang Yoon, Joo-Young Lee, Hae-Chang Rim and Soo-Hong Kim. A Practical QA System in Restricted Domains. In *Proceedings of 42nd Annual Meeting of the Association for Computational*

Linguistics, Workshop on Question Answering in Restricted Domains, Barcelona, Spain, July 2004.

[3] Jacques Courtin and Damien Genthial. Parsing with Dependency Relations and Robust Parsing. In *Proceedings of COLING-ACL '98 Workshop on Processing of Dependency -Based Grammars*, pages 88-94, Montreal, August 1998.

[4] Dina Demner-Fushman and Jimmy Lin. Knowledge Extraction for Clinical Question Answering: Preliminary Results. In *Proceedings of the AAAI-05 Workshop on Question Answering in Restricted Domains*, Pittsburgh, Pennsylvania, July 2005.

[5] John W Ely, Jerome A Osheroff, Paul N Gorman, Mark H Ebell, M Lee Chambliss, Eric A Pifer and P Zoe Stavri. A taxnomy of generic clinical questions: classification study. *BMJ 2000*, 321:429–432, 2000.

[6] S. Harabagiu, G.A. Miller, and D.I. Moldovan. WordNet 2 - A Morphologically and Semantically Enhanced Resource. In *Proceedings of ACL-SIGLEX99: Standardizing Lexical Resources*, Maryland, June 1999, pp.1-8.

[7] D. Lin. Dependency-based evaluation of minipar. In *Workshop on the Evaluation of Parsing Systems*, Granada, Spain, 1998.

[8] Donald A. B. Lindberg, Betsy L. Humphreys, and Alexa. T. McCray. The Unified Medical Language System. In *Methods of Information in Medicine, 32(4)*, pages 281-291, August 1993.

[9] Betsy L. Humphreys, and Donald A. B. Lindberg. The UMLS project: making the conceptual connection between users and the information they need. In *Bull Medical Libr Assoc.* 1993; 81:170-7.

[10] G.A. Miller WordNet: An on-line lexical database. *International Journal of Lexicography 3, 4* (Winter 1990), pp.235-312.

[11] Dan Moldovan and Vasile Rus. Logic Form Transformation of WordNet and its Applicability to Question-Answering. In *Proceedings of 39th Annual Meeting of the Association for Computational Linguistics*, Toulouse, France, July 2001.

[12] Dan Moldovan, Christine Clark, Sanda Harabagiu, and Steve Maiorano. COGEX: A Logic Prover for Question Answering. In *Proceedings of HLT-NAACL 2003. Human Language Technology Conference*, pages 87–93, Edmonton, Canada, 2003.

[13] Diego Mollá. Towards incremental semantic annotation. In *Proceedings of 1st International Workshop on Multimedia Annotation*, Tokyo, Japan, 2001.

[14] Diego Mollá, Rolf Schwitter, Michael Hess and Rachel Fournier. ExtrAns, an answer exraction system. *T.A.L. special issue on Information Retrieval oriented Natural Language Processing*, pages 495–522, 2002.

[15] Yun Niu, Graeme Hirst, Gregory McArthur and Patricia Rodriguez-Gianolli. Answering clincal questions with role identification. In *Proceedings of 41st annual meeting of the Association for Computational Linguistics, Workshop on Natural Language Processing in Biomedicine*, Sapporo, Japan, July 2003.

[16] Yun Niu and Graeme Hirst. Analysis of Semantic Classes in Medical Text for Question Answering. In *Proceedings of 42nd Annual Meeting of the Association for Computational Linguistics, Workshop on Question Answering in Restricted Domains*, Barcelona, Spain, July 2004.

[17] Elizabeth J. O'Neil, Patrick E. O'Neil and Gerhard Weikum. The LRU-K Page Replacement Algorithm For Database Disk Buffering. In *ACM SIGMOD Record*, Volume 22, Issue 2 (June 1993), pages 297–306.

[18] Fabio Rinaldi, James Dowdall, Gerold Schneider and Andreas Persidis. Answering Questions in the Genomics Domain. In *Proceedings of 42nd Annual Meeting of the Association for Computational Linguistics, Workshop on Question Answering in Restricted Domains*, Barcelona, Spain, July 2004.

[19] Yutaka Sasaki. Question Answering as Question-Biased Term Extraction: A New Approach toward Multilingual QA. In *Proceedings of 43th Annual Meeting of the Association for Computational Linguistics*, Michigan, USA, June 2005.

[20] Jose Luis Vicedo, Maximiliano Saiz, Ruben Izquierdo and Fernando Llopis. Does English Help Question Answering in Spanish. In *Proceedings of the 5th Workshop of the Cross-Language Evaluation Forum, CLEF 2004*, Bath, UK, September 2004.

[21] Ingrid Zukerman and Bhavani Raskutti. Lexical Query Paraphrasing for Document Retrieval. In Hsin-Hsi Chen and Chin-Yew Lin, editors, *Proceedings of the 19th International Conference on Computational Linguistics, COLING 2002*, Taipei, Taiwan, August 2002.